

Optimal Control of Satellite Attitude Acquisition  
by a Random Search Algorithm on a Hybrid Computer

by

William P. Kavanaugh, Elwood C. Stewart

and David H. Brocker

Ames Research Center, NASA

Moffett Field, California



For presentation at the 1968 Spring Joint Computer Conference, Atlantic  
City, New Jersey, April 30 to May 2, 1968.

FACILITY FORM 602

N 68 - 34394	
(ACCESSION NUMBER)	(THRU)
41	1
(PAGES)	(CODE)
TMX-61197	10
(NASA CR OR TMX OR AD NUMBER)	(CATEGORY)

- 2 -

ABSTRACT

Interesting and important control problems may be formulated as a problem of seeking the global minimum of a performance surface. Generally, the equations of this surface are unknown; further, the surface is often discontinuous, and has many peaks, valleys, and flat regions. Random search techniques are most appropriate in treating such problems. In a previous paper, an adaptive random search algorithm was developed and applied to the problem of seeking the minimum of a boundary surface associated with the two-point boundary-value problem that results from an application of the Maximum Principle. In the present paper, this same algorithm is reviewed and applied to a more complex problem: the minimum-fuel, large-angle, single-axis attitude acquisition problem. Comparison to optimal proportional control shows striking improvement in performance. Cross sections through the boundary surfaces reveal many of the irregularities noted above.

- 3 -

## INTRODUCTION

Computer implemented parameter search techniques for optimization problems have become useful engineering design tools over the past few years. Many, if not most of the techniques, are based on deterministic schemes which have inherent limitations when the system is nonlinear. Random search techniques have been suggested which propose to overcome some of the difficulties. References 1-3 give good general discussions of the merits of random techniques. Reference 4 develops an algorithm, based on random methods, to solve the difficult mixed two-point boundary value problem that results from an application of the Maximum Principle. The method was shown to be remarkably effective in solving a fairly complex fifth-order, nonlinear orbital-transfer problem. The purpose of this paper is to discuss the application of the random search algorithm to a still more complex problem to demonstrate its feasibility. The example chosen was the three-dimensional, large-angle, single-axis attitude acquisition control problem in which it is desired to minimize fuel expenditure to accomplish the acquisition. The equations are highly nonlinear since small angle assumptions cannot be made; the control torques are assumed to be limited. This problem is more complex than the orbit-transfer problem in that the dimension of the state vector (6) is greater by 1 and the number of degrees of freedom allowed the control action is greater. The same acquisition problem was discussed in reference 5 but a proportional control law was assumed. A random parameter search was used in that paper to find the optimal set

- 4 -

of feedback constants for the given control system structure so as to minimize system performance (fuel). Systems performances will be compared to indicate the striking improvement in performance with optimal nonlinear control.

In the following sections we will state the control problem for notational purposes, review the random search algorithm developed in detail in reference 4, discuss the hardware and software necessary for implementing the algorithm, and last, present the results of applying the method to the satellite acquisition problem.

#### PROBLEM FORMULATION

The problems considered are restricted to those for which the Maximum Principle is applicable. Although familiarity with the principle is assumed, a few remarks are necessary to properly pose the problems we will be concerned with in this paper. The system to be controlled is defined by the vector equation

$$\dot{x} = f(x, u, t) \quad (1)$$

where  $x = (x_1, x_2, \dots, x_n)$ ,  $u = (u_1, \dots, u_r)$  and  $u \in U$  where  $U$  is the allowable control region. Interest will center on fixed-time problems because of their convenience in computer operations. It will be desired to take the system from a given state  $x(0)$  to a final target set  $S$  so as to minimize the generalized cost function

$$C = \sum_{i=0}^n \alpha_i x_i(T) \quad (2)$$

- 5 -

where  $x_o(t)$  is the auxiliary state associated with the quantity to be minimized. The target set  $S$ , for the example chosen here, will be defined as  $S = x_f \in R_n$ , that is, a fixed point in  $n$ -dimensional space.

It is well known that application of the Maximum Principle to the stated problem invariably requires the solution to the following set of equations:

$$\left. \begin{aligned} u &= u(x, p, t) \\ \dot{x} &= f(x, u, t) \\ \dot{p} &= p(p, x, u, t) \end{aligned} \right\} \quad (3)$$

where  $p = (p_1, \dots, p_n)$ . We can see that the Maximum Principle yields a good deal of information about the nature of the control, that is, we know the function  $u(x, p, t)$ , and we know the equations for  $x$  and  $p$ . However, at no time do we know the specific values of both  $x$  and  $p$ . For example, at the initial time,  $x(0)$  is generally known from the problem specifications, whereas  $p(0)$  is not known. The remaining boundary conditions required will be known at the final time by some combination of components from the  $x$  and  $p$  vectors. Thus, there is difficulty in solving these equations even numerically because the known boundary conditions are split between initial and final times.

#### RANDOM SEARCH ALGORITHM

In this paper we will use the algorithm developed in reference 4 for solving the mixed boundary-value problem. Consequently, we will give only an intuitive account of this approach necessary for the later example application.


A direct way of solving the mixed boundary-value problem is to convert it into an initial-value problem. From equations (3) it is clear that for any arbitrary value of  $p(0)$ , there will be sufficient information to determine a final state  $x(T)$ . Since this state will generally be different from the desired state  $x_f(T)$ , we introduce a vector metric  $J$  (see ref. 4 for the significance of using a vector metric rather than a scalar metric) to measure the distance between  $x(T)$  and  $x_f(T)$ . It is convenient conceptually to think of the components of the vector quantity  $J$  as hypersurfaces in an  $n$ -dimensional space of the components of the  $p$  vector. Then the boundary-value problem is equivalent to finding the simultaneous minima of all the hypersurfaces. It is important to note that in this case the minimum values are known, i.e., zero.

Deterministic approaches for finding the minima of the hypersurfaces have a number of difficulties. For example, the gradient technique requires the calculation, or possible experimental measurement, of the partial derivatives of the surfaces at each step of an iteration process. If the surfaces are discontinuous, have many relatively rapid slope changes, or regions of zero slope, gradient

- 7 -

approaches will fail. The random search techniques overcome these difficulties. References 1 through 4 discuss the virtues of these methods in greater detail. In particular, it is demonstrated in reference 4 that many of the hypersurface abnormalities mentioned above actually occur in even a moderately complex problem.

The random search approach to be used here was described in considerable detail in reference 4. The approach is based on a direct search of the hypersurfaces by selecting the initial condition vector  $p(0)$  from a gaussian noise source, followed by an evaluation of the corresponding values of the hypersurfaces. It was shown that the pure random search is not practical for moderately high-order systems because of the slow convergence to the minimum. However, by making the search algorithm adaptive, the convergence properties were shown to be greatly improved. This was accomplished by: (a) varying the mean value of the gaussian distribution on any iteration so as to equal the initial condition of the adjoint vector on the last successful iteration, and (b) varying the variance of the distribution so that the search is localized when the iterations are successful but gradually expanded in a geometric progression when not successful. Thus, the mean provides a creeping and direction-seeking character to the search while the variance provides an expanding and contracting character.

The behavior of the algorithm is illustrated in figure 1 by the  typical boundary function surface given in only two dimensions.

- 8 -

Starting at point 1, the mean of the distribution is made equal to the corresponding value of  $p(o)$ ; the search starts with the small variance indicated and gradually expands in geometrical steps until a lower point on the surface is detected, such as at point 2. Then the mean of the distribution is made equal to the corresponding value of  $p(o)$  and the search repeats. The search continues as indicated by the typical numbered points until a value of zero or some small value,  $\epsilon$ , near zero is reached.

This type algorithm has some desirable properties that enable it to find the minimum. For example, it has a local minimum-seeking property that is due to the small variance used on those iterations which are successful. The algorithm also has a global searching property that enables it to jump over peaks, which is due to the expansion of the variance when the iterations are not successful. Further, it will not matter whether the surface is discontinuous, or has many peaks, valleys or flat regions.

#### IMPLEMENTATION


The hybrid computer proved to be the most feasible way to implement the random search algorithm. A primary reason for this is that a relatively large number of iterations are required to find a solution. Reference 4 showed that approximately 8000 iterations were required on the average for a typical solution. Each iteration can be, from a computational point of view, divided into two steps: (1) integration



- 9 -

of the equations of motion on the interval  $[0, T]$ , and (2) execution of algorithm logic. The analog computer is by far the faster machine in performing step one. Although the second step might be accomplished in approximately the same time with either machine it is best done digitally. Thus the conclusion is reached that a hybrid approach requires a great deal less computer time than a completely digital simulation. It is worth noting that an alternative approach with pseudo-hybrid techniques was investigated using an analog computer and something less than a digital computer. However, our experience shows that inaccuracies, limited storage and limited flexibilities in logical operations seriously limit the feasibility of this approach.

In the hybrid implementation, the analog computer was delegated the task of solving the state, adjoint, and control equations as given in equation (3). It also served as the point at which the operator exercised manual control over the hybrid system. The digital computer was required to calculate the metric, provide storage, implement the algorithm logic, randomly generate the initial conditions for the adjoint equations and, finally, oversee the sequencing of events of the iterate cycle. This division of computational effort is shown schematically in figure 2 which is a block diagram of the search algorithm. The superscript  $k$  shown in this figure designates the generic  $k^{\text{th}}$  iteration of a long sequence of iterations. For convenience the flow of information through the block diagram may be thought to start at  $p(0)$  which represents the adjoint initial conditions in analog

Fig. 2

- 10 -

form. They are applied to their respective initial condition circuits on the analog machine, and then that machine is commanded into an operate mode. The set of equations (3) are solved on the time interval  $[0, T]$ , and at time  $T$  those analog components we are interested in reading are commanded into a hold mode and the executive sequencing program instructs the A/D converters to read these variables. On the basis of this information the boundary cost function metric is computed digitally, and then tested by comparing it to the last smallest value discussed above. On the basis of this test information, the algorithm logic operates to control the mean value  $m^k$  and variance  $\sigma^k$ . A new random vector  $p^k = m^k + \xi^k$  is then generated, converted to an analog signal, and applied to the  $p(0)$  initial condition circuits. The whole process is repeated for the next iteration.

Figure 3 is a hardware diagram of the hybrid system used. Shown are the two basic elements of the simulation, the analog and digital computers along with their coupling system, and peripherals. The coupling system is comprised of two distinct parts: (a) the Linkage Systems and (b) the Control Interface System. A discussion of the hardware used in these subsystems is given in the four sections to follow. The next (fifth) section discusses the sequencing of events through the subsystems during one iteration cycle in order to better describe the functioning of the hybrid system as a whole. Discussed in the final section is the flow graph for the algorithm.

Fig. 3

- 11 -

### Digital Computer

The digital computer used in the optimization program was an Electronic Associates, Inc. (EAI) model 8400. The particular machine used has 16,000 words of core memory with 32 bits per word. Memory cycle time is 2 microseconds. The machine uses parallel operation for maximum speed. Floating point operations are hardware implemented. The optimization program was coded in MACRO ASSEMBLY in order to keep the execution time to a minimum. The instruction repertoire includes special commands by which discrete signals can be sent to or received from the external world. External interrupts are provided which can trap the computer to a specific cell in memory. In an example to be discussed later, the optimization program utilized about 8,000 words of storage. Of these about 1,000 comprised the actual optimization executive program, the remaining 7,000 being used for subroutines, monitor and on-line debugging and program modification routines.

### Analog Computer

The analog hardware consisted of an Electronic Associates 231R-V analog computer. Since the state equations, adjoint equations, and the control logic were programmed in standard fashion, analog schematics were not included.

The analog computer serves as the point at which mode control of the hybrid computer is accomplished. By manual selection of switches either of two modes can be commanded: (1) In the "search" mode the

- 12 -

analog computer operates in a high-speed repetitive manner. Such operation is accomplished by controlling the mode of the individual integrators with an appropriate discrete signal. This signal is a two-level signal which is generated on the control interface in conjunction with the digital computer and, depending on the level, holds an integrator in either "operate" or "initial condition" mode. (2) In the "reset" mode, the integrators are placed in their initial-condition mode and held there.

For continuous type output, a display console was connected to the analog computer to provide visual readout of variables. The display contained a cathod ray tube (CRT) which could simultaneously display up to four channels, and enabled photographic records to be taken of the display quantities. The display was extremely helpful in determining if the algorithm was functioning properly.

#### Control Interface

The control interface between the analog and digital system is an Electronic Associates, Inc. DOS 350 (see fig. 3). It is through this unit that the iteration process is controlled. An important task allocated to this subsystem is the operate-time control. This function is implemented through the use of a counter and is the key element in the control of all timing in the hybrid simulation. The counter is driven from a high-frequency source in the interface system allowing for a very high degree of resolution in the simulated operate-time. Also, the

- 13 -

interface allows the digital computer to use any conditions in the analog computer which can be represented by discrete variables (binary levels) and to send discrete signals to the analog system to be used as control levels or indicators. As example of the former would be the hybrid system mode control which merely amounted to the operator depressing the "reset" or "search" switch on the analog computer. This action sets a binary level which is then sensed by the digital computer. An example of the latter situation is when the digital sends the operate command to the operate-time counter. The interface system allows patching of Boolean functions. Hence, some of the logic operations required for timing pulses, event signals, and other like operations were very effectively programmed on it.

#### Linkage System

The linkage system shown in figure 3 houses the conversion equipment, the A/D and D/A converters. It is through here that all of the data pass between the analog and digital portions of the simulation. The linkage system is controlled by command from the digital computer.

Input to the digital computer is through the A/D converter via a channel selection device or multiplexer that selects the analog channel to be converted. Conversions were sequential through the analog channels at a maximum rate of 80,000 samples per second from channel to channel.

Output to the analog used the D/A converters, with each data channel having its own conversion unit. The maximum conversion rate of the D/A's

- 14 -

used in 250,000 conversions per second.

#### Sequencing of Events During One Iteration

The sequencing of events during one iteration cycle are depicted in figure 4. The instants of time  $t_1, t_2, \dots, t_5$  shown in this figure are considered fixed relative to each other, and  $t_1$  is conveniently regarded to be the start of the iteration cycle. We will consider the cycle to begin at  $t_1$  with the analog integrators in an operate mode. As discussed previously, the elapsed time  $(t_2 - t_1)$  is controlled by a counter on the interface system. At  $t_2$  an interrupt pulse is generated on the control interface which is sent to the digital computer signaling it to commence its operations. Simultaneously, the pulse is sent to the analog to instruct the track-store units to hold their respective values which they had at time  $t_2$ . During the interval  $(t_3 - t_2)$  the digital computer reads these analog variables with the A/D converter. At  $t_3$  the digital sends a pulse via the interface to the analog console which commands the integrators to an initial condition mode. At  $t_4$ , when the data required by the algorithm have been generated, the D/A converters send these values to the appropriate points in the analog portion of the simulation. The digital machine allows enough time for the transients to settle in the initial condition circuits of the analog before sending a command at  $t_5$  that places the integrators in an operate mode and starts the counter. Since  $t_5$  and  $t_1$  are the same event, we merely repeat the above sequence for repetitive operation.

Fig. 4

- 15 -

Some specific numerical values might be of interest. The total iterate time  $(t_5 - t_1)$  is primarily composed of two parts: (1)  $(t_2 - t_1)$  which in a later example problem was scaled in the simulation to 7.5 milliseconds, and (2)  $(t_5 - t_2)$ , which was primarily determined by the speed of the digital machine in computing, converting, and generating random numbers; this latter period was on the order of 7.5 milliseconds. Thus, the total iterate time for the above situation is on the order of 15 milliseconds (or 66 iterations per second). This figure is dependent on the control problem chosen and the exact form of the algorithm implemented.

#### Algorithm Flow Graph

Figure 5 is a program flow graph showing the software requirements of the iteration process. This basically constitutes a majority of the steps involved in the algorithm and the iteration control sequences utilized by the hybrid system. Note the inclusion of the event times  $t_1, t_2, \dots$  discussed earlier in connection with figure 4. The program is continuously recycling in a high-speed repetitive fashion.

&lt; Fig. 5

There are three basic loops in figure 5 corresponding to the three system modes in the optimization program: a reset loop, a search loop, and an end-state loop. The reset loop initializes the program. The search loop uses the algorithm to search for a solution to the problem. The end-state loop is entered by the digital program when a solution is found, and is used for generating graphic displays. The operator

- 16 -

manually selects the search or reset mode as discussed in the section dealing with the analog computer. A more detailed description of these loops is given in reference 4.

## APPLICATION

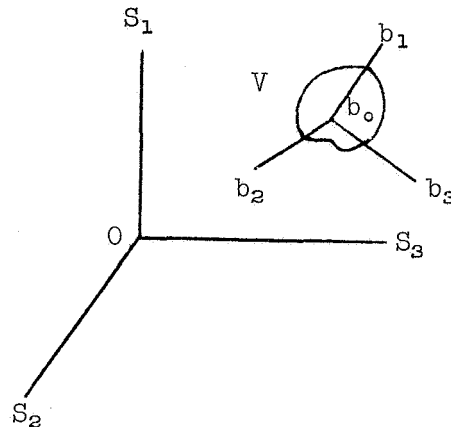
In this section we will discuss the application of the random search algorithm to the single-axis attitude acquisition control problem. In the following we will first formulate the problem by giving a physical description of the problem and writing out the exact equations of motion. Second, we will outline the equations necessary for determining optimal nonlinear control as derived by means of the Maximum Principle; for comparative purposes we will also outline the optimal proportional control derived in reference 5. Next, we will discuss the boundary conditions and the vector metric. In the final two subsections we will illustrate the computer results: in one we will give a variety of time history solution and fuel performances, and in the other, some cross sections through the boundary cost-function hypersurfaces.

### Formulation

Consider a freely rotating vehicle  $V$  about a point  $b_0$  in inertial space defined by the set of axes  $(S_1, S_2, S_3)$  shown in the sketch.



- 17 -



A fixed set of body axes,  $b_1, b_2, b_3$ , is ascribed to the vehicle in the principle axes of inertia with origin at  $b_0$ . The orientation of any body axis with respect to inertial space will be specified by direction cosines, e.g.,  $a_{13}, a_{23}, a_{33}$  where  $a_{13}$  is the cosine of the plane angle between  $S_1$  and  $b_3$ , and similarly for  $a_{23}$  and  $a_{33}$ . The orientation of the vehicle is specified by a  $(3 \times 3)$  direction cosine matrix. Since we are interested in single-axis orientation, we will only require knowing three direction cosines instead of nine. In the study we will orient  $b_3$  in inertial space. The control torques required to orient the vehicle are produced by mass expulsion devices aligned with each of the three axes. It is assumed the mass flow rates are used to vary the torques, and that they are bounded (except when examining proportional control). The vehicle is inertially unsymmetrical and is considered to be in a general tumbling motion at the initial time. The objective of control is to apply torques for a fixed period of time in a manner that will reduce total momentum to zero and orient

- 18 -

the  $b_3$  axis of the vehicle from any initial orientation to any other prescribed orientation in inertial space. Furthermore, we must accomplish this task with the control program that uses the least amount of fuel. This verbal statement of the problem will now be formulated more explicitly.

The dynamical equations of motion of a body of fixed inertia rotating about a point in inertial space and acted upon by external torques are given by the following set of equations:

$$\left. \begin{aligned} \dot{\omega}_1 &= \{(1-\beta)/\alpha\} \omega_2 \omega_3 + \gamma_1/\alpha \\ \dot{\omega}_2 &= \{(\beta-\alpha)\} \omega_3 \omega_1 + \gamma_2 \\ \dot{\omega}_3 &= \{(\alpha-1)/\beta\} \omega_1 \omega_2 + \gamma_3/\beta \end{aligned} \right\} \quad (4)$$

where

$$\begin{aligned} \alpha &= I_1/I_2 && \text{Roll to pitch inertia ratio} \\ \beta &= I_3/I_2 && \text{Yaw to pitch inertia ratio} \\ \left. \begin{aligned} \gamma_1 &= M_1/I_2 \\ \gamma_2 &= M_2/I_2 \\ \gamma_3 &= M_3/I_2 \end{aligned} \right\} && \begin{aligned} &\text{Control accelerations, rad/sec}^2 \\ &\text{normalized to } I_2 \end{aligned} \\ \omega_i &= \text{body rates, rad/sec} && i = 1, 2, 3 \\ M_i &= \text{torque, lb/ft} \end{aligned}$$

The three kinematical variables (direction cosines) required to specify the orientation of  $b_3$  are designated  $a_{13}$ ,  $a_{23}$ ,  $a_{33}$ . These variables are related to the dynamical variables by the following set of differential equations (see ref. 5 for a discussion on this):

- 19 -

$$\left. \begin{aligned} \dot{a}_{13} &= \omega_3 a_{23} - \omega_2 a_{23} \\ \dot{a}_{23} &= \omega_1 a_{33} - \omega_2 a_{13} \\ \dot{a}_{33} &= \omega_2 a_{13} - \omega_1 a_{23} \end{aligned} \right\} \quad (5)$$

For this study the specific values of the roll and pitch inertia ratios were taken to be  $\alpha = 1.15$ ,  $\beta = 0.48$ . Also, the maximum control torque acceleration permitted in the nonlinear controller situation was limited to approximately one-sixth the peak-acceleration required for proportional control, or  $\gamma_{1_{\max}} = \gamma_{2_{\max}} = 2\gamma_{3_{\max}} = 7.5 \text{ rad/sec}^2$ .

To transfer (4) and (5) into state variable form, the following substitutions are made:

$$\begin{aligned} a_{13} &= x_1 & \omega_1 &= x_4 & u_1 &= \gamma_1/\alpha \\ a_{23} &= x_2 & \omega_2 &= x_5 & u_2 &= \gamma_2 \\ a_{33} &= x_3 & \omega_3 &= x_6 & u_3 &= \gamma_3/\beta \end{aligned}$$

This gives us the following set of state variable equations:

$$\left. \begin{aligned} \dot{x}_1 &= x_6 x_2 - x_5 x_3 \\ \dot{x}_2 &= x_4 x_3 - x_6 x_1 \\ \dot{x}_3 &= x_5 x_1 - x_4 x_2 \\ \dot{x}_4 &= \{(1-\beta)/\alpha\} x_5 x_6 + u_1 \\ \dot{x}_5 &= \{(\beta-\alpha)\} x_6 x_4 + u_2 \\ \dot{x}_6 &= \{(\alpha-1)/\beta\} x_4 x_5 + u_3 \end{aligned} \right\} \quad (6)$$

The objective of the control is to take the state vector from an arbitrary initial value  $x(0)$  to an arbitrary final value  $x_f(T)$  in a fixed interval of time  $[0, T]$ , and use the least amount of fuel in so doing. A new coordinate, proportional to the total fuel used in all three

- 20 -

axes, can be defined as follows:

$$x_o(t) = \int_0^t \sum_{i=1}^3 |u_i(\tau)| d\tau \quad (7)$$

and we can then interpret the objective as the minimization of the terminal value  $x_o(T)$ .

### Control Laws

The nonlinear optimal control can be derived by an application of the Maximum Principle. This derivation will not be given here but a summary of the equations necessary for computer implementation will be given. First are the adjoint equations which can be shown to be:

$$\left. \begin{aligned} \dot{p}_1 &= p_2 x_6 - p_3 x_5 \\ \dot{p}_2 &= p_3 x_4 - p_1 x_6 \\ \dot{p}_3 &= p_1 x_5 - p_2 x_4 \\ \dot{p}_4 &= p_3 x_2 - p_2 x_3 - p_5 x_6 (\beta - \alpha) - p_6 x_5 (\alpha - 1) / \beta \\ \dot{p}_5 &= p_1 x_3 - p_3 x_1 - p_4 x_6 (1 - \beta) / \alpha - p_6 x_4 (\alpha - 1) / \beta \\ \dot{p}_6 &= p_2 x_1 - p_1 x_2 - p_4 x_5 (1 - \beta) / \alpha - p_5 x_4 (\beta - \alpha) \end{aligned} \right\} \quad (8)$$

Second are the equations defining the optimal control vector at each instant of time:

$$\begin{aligned} u_i(t) &= N_i \operatorname{sgn} p_{i+3}(t) & \text{if } |p_{i+3}(t)| > 1 \\ u_i(t) &= 0 & \text{if } |p_{i+3}(t)| < 1 \end{aligned} \quad (9)$$

where  $i = 1, 2, 3$  and  $N_i$  is the maximum torque acceleration allowed in the  $i^{\text{th}}$  control axis. It is seen that the control torque is of

- 21 -

the on-off character and that torque direction is obtained by assigning the correct sign to the "on" signal according to equation (9).

An optimal proportional control law used in this paper for comparative purposes was taken from reference 5 and is discussed briefly here for the sake of completeness. In reference 5 the structure of the optimal control law was assumed to be of the form:

$$\left. \begin{aligned} u_1 &= -Dx_4 - Ex_2 \\ u_2 &= -Fx_1 + Gx_1 \\ u_3 &= -Hx_6 \end{aligned} \right\} \quad (10)$$

The parameters D, E, F, G and H were left free, and by means of a random parameter search suitable values were found for which the stated objective of the problem (zero momentum and alinement of  $b_3$  to  $S_3$ ) was achieved. The search was repeated a number of times, each time observing system performance given by equation (7). An optimum parameter vector was selected from the set of parameter vectors which satisfied the problem objective and minimized the performance criteria. This parameter vector, in conjunction with equation (10), defines optimal proportional control. This control may be difficult to achieve in practice, however, since no bounds have been imposed on the thrust. When there are bounds, the control law is then referred to as optimal saturating proportional control.

- 22 -

The Vector Metric

The desired boundary condition at the terminal time was chosen to be zero momentum and alignment of the body axis  $b_3$  with the inertial axis  $S_3$ ; this is expressed by  $x_f(T) = (0, 0, 1, 0, 0, 0)$ . To satisfy these boundary conditions, it is necessary in the random search approach to introduce the vector metric  $J$  as discussed previously. Its general form was specified in reference 4 to be  $J = (J_D, J_V, J_P)$  where the subscripts on the components refer to displacement, velocity, and adjoint variables, respectively. However, in this application, since all terminal states  $x_f(T)$  are fixed,  $p(T)$  is completely free so that we may ignore the  $J_P$  component in the vector metric. For the present example  $J_D$  and  $J_V$  are taken to be

$$J_D = \sqrt{x_1^2 + x_2^2 + (x_3 - 1)^2}$$

$$J_V = \sqrt{x_4^2 + x_5^2 + x_6^2}$$

It is clear that  $J_V = 0$  implies  $x_4, x_5, x_6 = 0$  which represents zero momentum as desired. Also,  $J_D = 0$  implies the desired final orientation  $x_1 = x_2 = 0$  and  $x_3 = 1$ . In actual practice we will only require

$$(J_D, J_V) < (\epsilon_D, \epsilon_V)$$

where the  $\epsilon$  values are chosen to meet the problem requirements. For the specific problem discussed below, the value of  $\epsilon_V$  chosen reduced a  $10^\circ/\text{sec}$  initial velocity error in each axis to approximately  $0.75^\circ/\text{sec}$  in

- 23 -

each axis at time  $T$ . The  $\epsilon_D$  chosen required that the  $b_3$  body axis be oriented to within a few degrees of the  $S_3$  inertial axis, from any initial orientation.

### Time History Solutions

In this section we will give some computer solutions for the satellite attitude acquisition problem. Our interest will center on results for the optimal nonlinear control obtained by implementing the random search algorithm discussed in the preceding sections. For comparison, we will also give results for no control and the proportional control studied in reference 5.

Table I summarizes the controllers to be studied, the initial conditions of the vehicle, and the resulting fuel requirements. It is worth emphasizing again that in the initial condition vector the first three components are the initial angular positions given in terms of direction cosines varying between -1 and +1. The latter three components are the initial angular velocities in degrees/second, and were chosen somewhat arbitrarily. The desired final condition vector is taken to be  $x_f(T) = (0,0,1,0,0,0)$ ; thus, the initial momentum is reduced to zero and the body axis  $b_3$  is to be kept aligned with the inertial axis  $S_3$ . Also, a solution is given for the more general case where initial misalignment exists.

Table I

A. No control - In figures 6(a) and (b) are shown the time histories of the state variables describing the motion of the system when no control is used and the vehicle starts with the initial condition

Fig. 6

- 24 -

$x(0) = (0,0,1,-10,10,10)$ . This corresponds to initial alinement of axes but with the initial angular velocities indicated. It can be noted that the angular positions vary over their entire range  $(-1,1)$  during the time interval  $[0,T]$ . From these results it might be anticipated that with limited torque control, the wide range of angular position variations would persist over a substantial portion of the interval. In this event, the equations of motion are distinctly nonlinear and it is inappropriate to linearize them. The Maximum Principle allows these nonlinearities to be dealt with directly.

B. Optimal Proportional Control - The time history solutions for the optimal proportional control derived in reference 5 (as discussed above) are given in figure 7 for the same initial condition as with no control (see Table I). As is well known, optimizing system performance under the assumption of proportional control often leads to impulsive-type control. This tendency can be seen in figure 7(c) where the initial torque acceleration rises to  $44 \text{ mrad/sec}^2$  and then decreases relatively quickly to zero. A normalized fuel consumption as measured by the computer was  $x_0(T) = 0.65$ . This is nearly two and one half times the minimum theoretical value of 0.28 obtained by allowing an ideal impulse of torque.

Fig. 7

In practice, the fuel requirements of 0.65 could only be attained with proportional control, i.e., without saturation of the torque. It was demonstrated in reference 5 that the effect of saturation on the proportional controller is to increase both the consumed fuel and the time



- 25 -

required to accomplish the mission. However, the results given there were for initial velocities of the vehicle of only  $4^\circ/\text{sec}$  rather than the  $10^\circ/\text{sec}$  used herein.

C. Optimal Nonlinear Control - The solutions for the optimal nonlinear control obtained by implementing the random search algorithm are given in figures 8 and 9. These figures correspond, respectively, to the cases indicated in Table I: initial alinement and no initial alinement.

Figs. 8  
and 9

The results given in figure 8 are for initial alinement of axes in which the initial condition vector is  $x(0) = (0,0,1,-10,10,10)$ , the same as for the other controllers discussed above. The solutions obtained are quite different from the previous cases as can be seen by comparing figure 8 to figures 6 and 7. It is noted that the maximum torque for optimal nonlinear control is approximately one-sixth the peak value for proportional control (note the different origins for the three control functions). From careful examination of the time histories, it appears that the optimal control law is anticipatory, that is, it appears to act at the most advantageous moment in order to reduce the large excursions of the states to the desired end values. Fuel consumption was found to be  $x_6(T) = 0.31$  which is only 1.1 times larger than for the optimal ideal impulse case. By comparison, the optimal proportional control is very inefficient. Proportional control uses 2.1 times the fuel for optimal nonlinear control. This factor would be even larger if the comparison were made to the more practical

- 26 -

saturating proportional control system. Note also in figure 8 that the operate times are nearly the same as for the proportional control system. Thus, we need not necessarily expect a time penalty for the practical constraint imposed by control saturation.

In the results given in figure 9, the initial condition vector  $x(o)$  is not aligned. The initial value is  $x(o) = (0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, -10, 10, 10)$  and corresponds to an initial rotation of  $b_3$  of  $45^\circ$  in the  $S_2, S_3$  plane. The time histories and optimal control thrust are seen to be similar to those with initial alignment. As for the fuel, we might expect that angular position error at the initial time would require a higher fuel consumption to obtain the same objectives. Indeed, experimental data show that fuel required is 0.43 which is 40 percent more than when the axes are initially aligned.

#### Boundary Cost Function Surfaces

It was pointed out previously that irregularity in the boundary cost function hypersurfaces reflects the need for a search algorithm that incorporates global as well as local-seeking properties. The highly irregular nature of these hypersurfaces for the attitude control problem is demonstrated in figure 10. Here are shown typical two-dimensional cross sections through the hypersurfaces at a solution point. The curves were obtained by slowly varying one of the adjoint variables from -100 volts to +100 volts while all others were fixed at their respective solution values. Note in particular the irregular

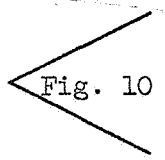


Fig. 10

- 27 -

multipeak nature of the surfaces and the occasional noisy appearance. It is also worth noting the rather narrow valleys surrounding the optimum point. These surfaces clearly indicate the difficulties a deterministic method might encounter. For example, the gradient method would have ample opportunity of "hanging up" in the wrong valley. Furthermore, the two-dimensional curves give only a hint of the difficulties to be expected in the actual six-dimensional space.

#### CONCLUDING REMARKS

It was demonstrated in reference 4 that the random search technique gave a practical approach to generating explicit optimal solutions for a moderately complex example problem under a wide range of conditions. An objective of the present study was to demonstrate the feasibility of using the random search technique in a different and more complex situation. The results presented in this paper gave a positive indication of that feasibility and together with reference 4 suggest that possibly many high-order nonlinear problems will have complicated mappings and that the random search methods would be highly successful in coping with them. Furthermore, the results of this paper show quite explicitly what gains in system performance can be made in a given situation by using rigorously derived optimal control compared to a more conventional "optimal" control procedure. Such knowledge gives greater impetus to seek efficient ways of implementing the optimal nonlinear control. The open-loop aspect of the controls generated in this manner

- 28 -

still limits any on-line capability. However, it is believed that off-line preliminary design capability for studying moderately complex nonlinear dynamical systems could be enhanced by this approach.

REFERENCES

1. Dean C. Karnopp  
Random Search Techniques for Optimization Problems  
Automatica, Vol 1, Pergamon Press, 1963, pp 111-121.
2. I. N. Bocharov and A. A. Feldbaum  
Automatic Optimizer For The Search For The Smallest of Several Minima  
Avtomatika i Telemekhanika, Vol 23, No 3, March 1962.
3. R. R. Favreau and R. G. Franks  
Statistical Optimization  
Proceedings 2nd International Analog Computer Conference, 1958.
4. E. C. Stewart, W. P. Kavanaugh, and D. H. Brocker  
Study of a Global Search Algorithm for Optimal Control  
Fifth Congress of International Association for Analog Computation,  
Lausanne, Switzerland, August 1967,
5. A. Sabroff, R. Farrenkopf, A. Frew, and M. Gran  
Investigation of The Acquisition Problem in Satellite Attitude Control  
Wright-Patterson Technical Report AFFDL-TR-65-115, December 1965.

TABLE I - COMPARISON OF CONTROL SYSTEMS

	<u>INITIAL CONDITION</u>	<u>FUEL</u>
No Control	(0,0,1,-10,10,10)	0
Optimal Proportional Control (no saturation)	(0,0,1,-10,10,10)	.65
Optimal Impulse Control	(0,0,1,-10,10,10)	.28
Optimal Nonlinear Control (with saturation)		
a. Initial alinement	(0,0,1,-10,10,10)	.31
b. Initial nonalinement	$(0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, -10, 10, 10)$	.43

FIGURE LEGENDS

- Figure 1. - Typical Boundary Cost Function Surface
- Figure 2. - Hybrid System Block Diagram for Random Search Method
- Figure 3. - Hybrid System Hardware
- Figure 4. - Sequencing of Events During One Iteration
- Figure 5. - Algorithm Flow Graph
- Figure 6. - Time History of State Vector; No Control; Initial Alinement of Axes
- Figure 7. - Time Histories of State and Control Vectors; Optimal Proportional Control; Initial Alinement of Axes
- Figure 8. - Time Histories of State and Control Vectors; Optimal Nonlinear Control; Initial Alinement of Axes
- Figure 9. - Time Histories of State and Control Vectors; Optimal Nonlinear Control; Initial Nonalinement of Axes
- Figure 10. - Example Two-Dimensional Cross Sections of Boundary Hyper-surfaces

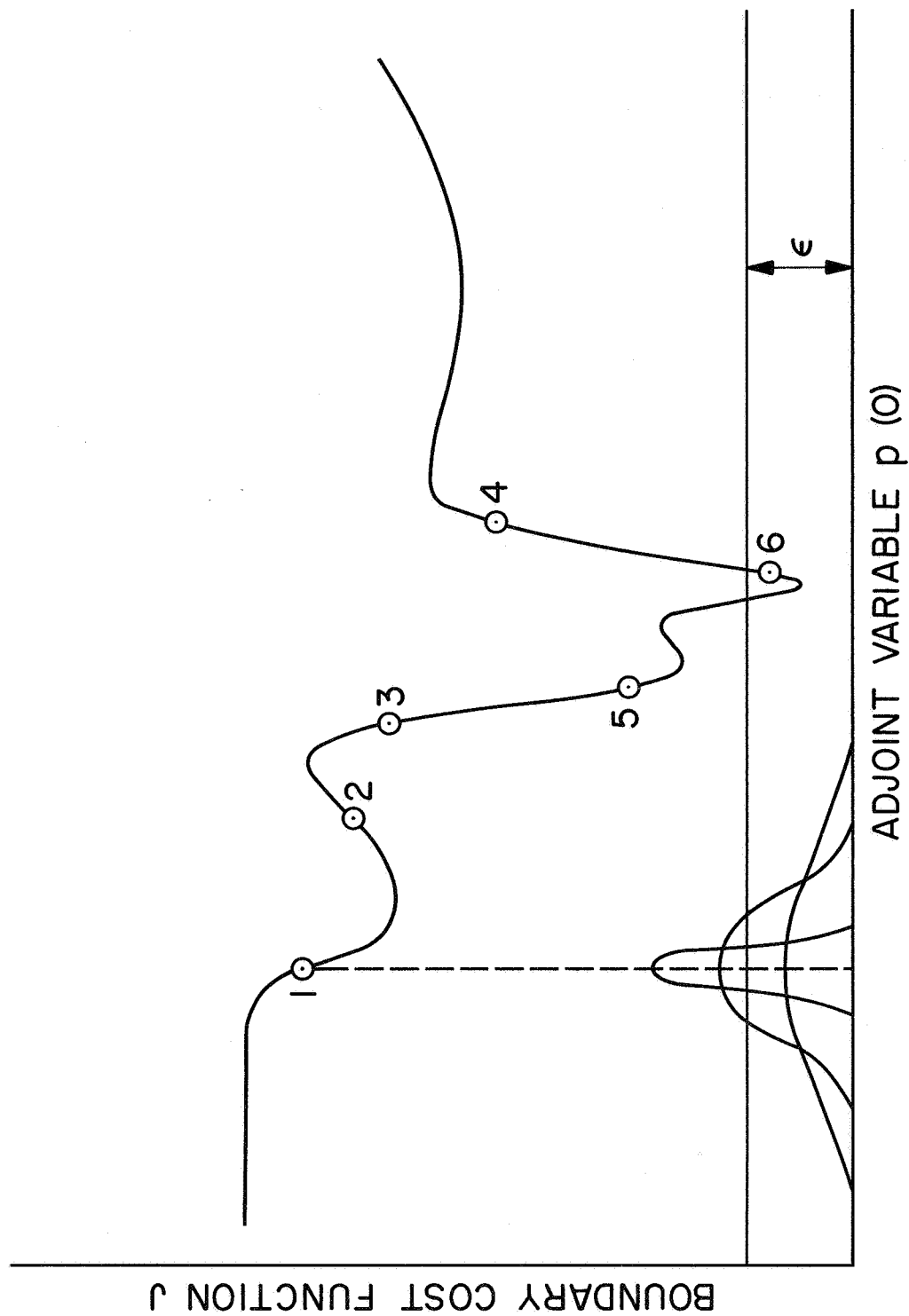


Figure 1.- Typical boundary cost function surface.



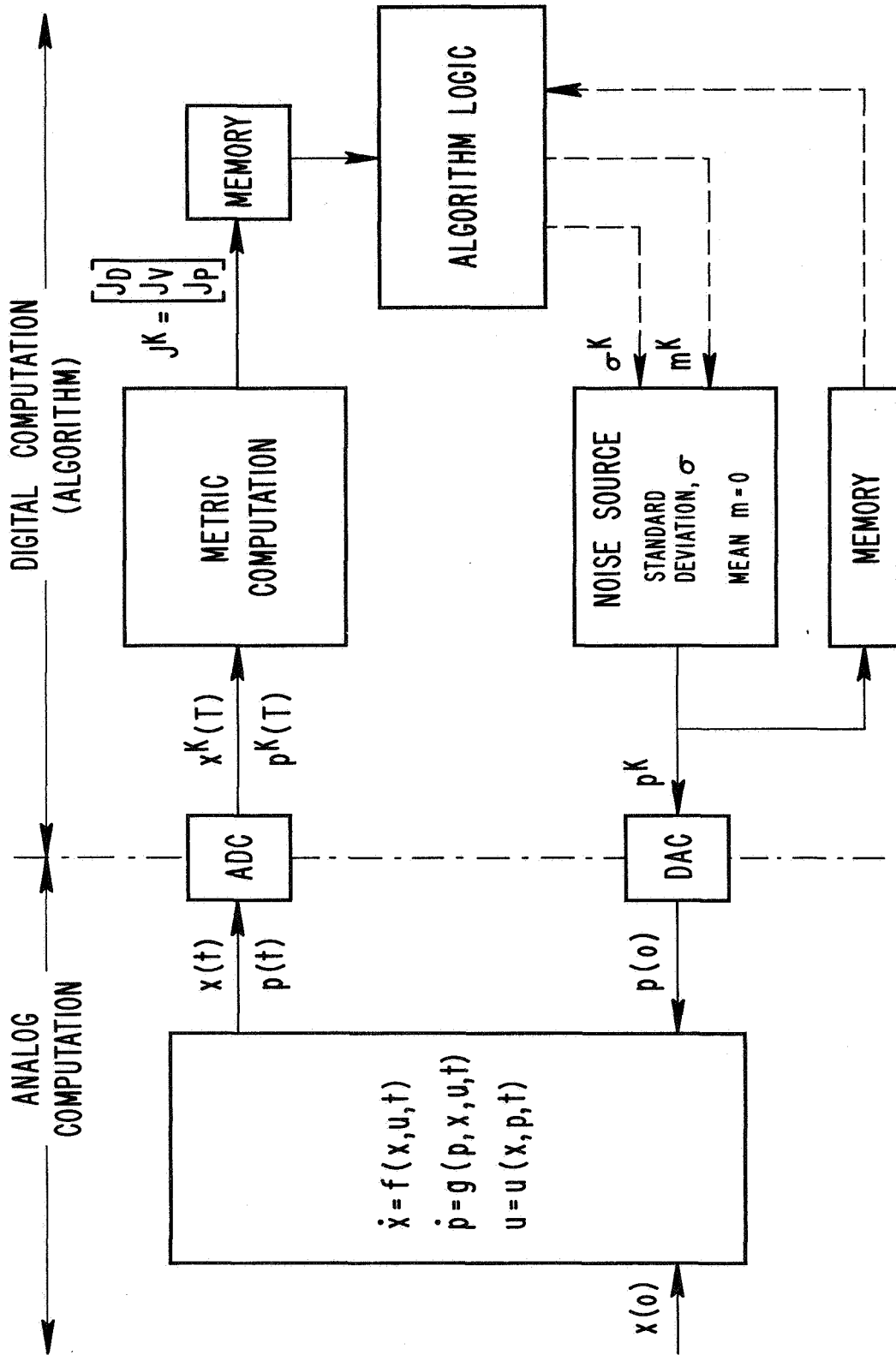


Figure 2.- Hybrid system block diagram for random search method.

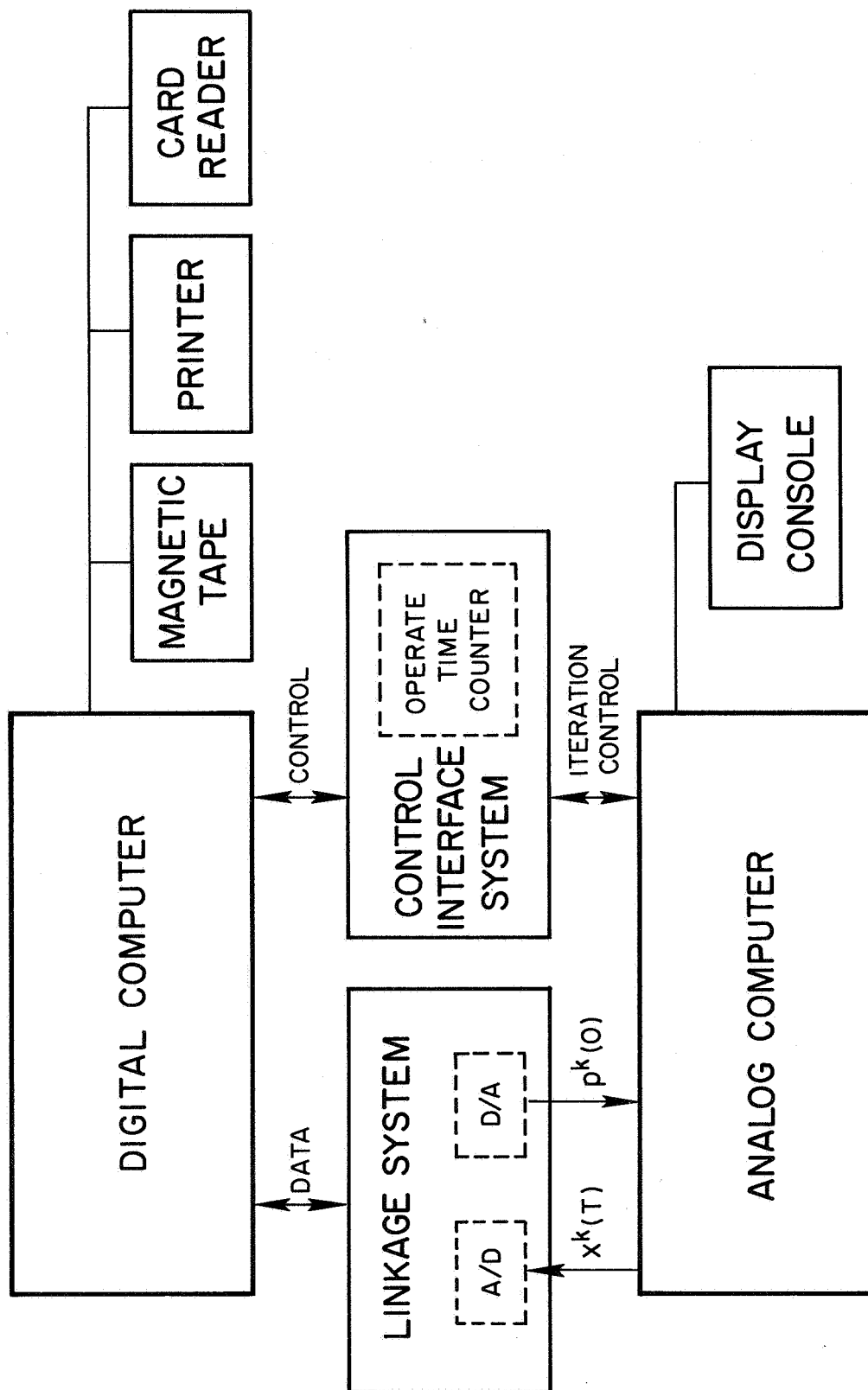


Figure 3.- Hybrid system hardware.

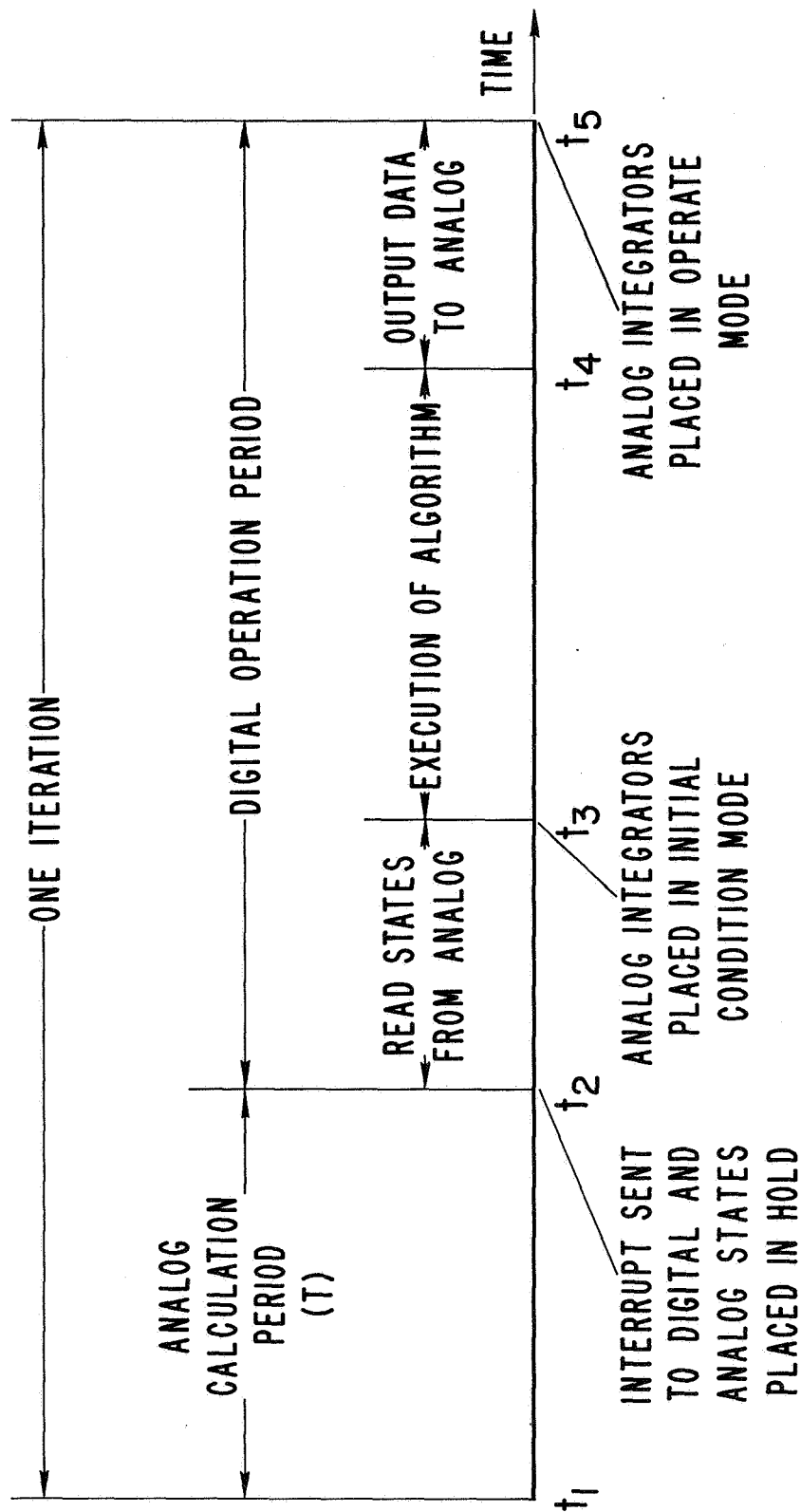


Figure 4.- Sequencing of events during one iteration.

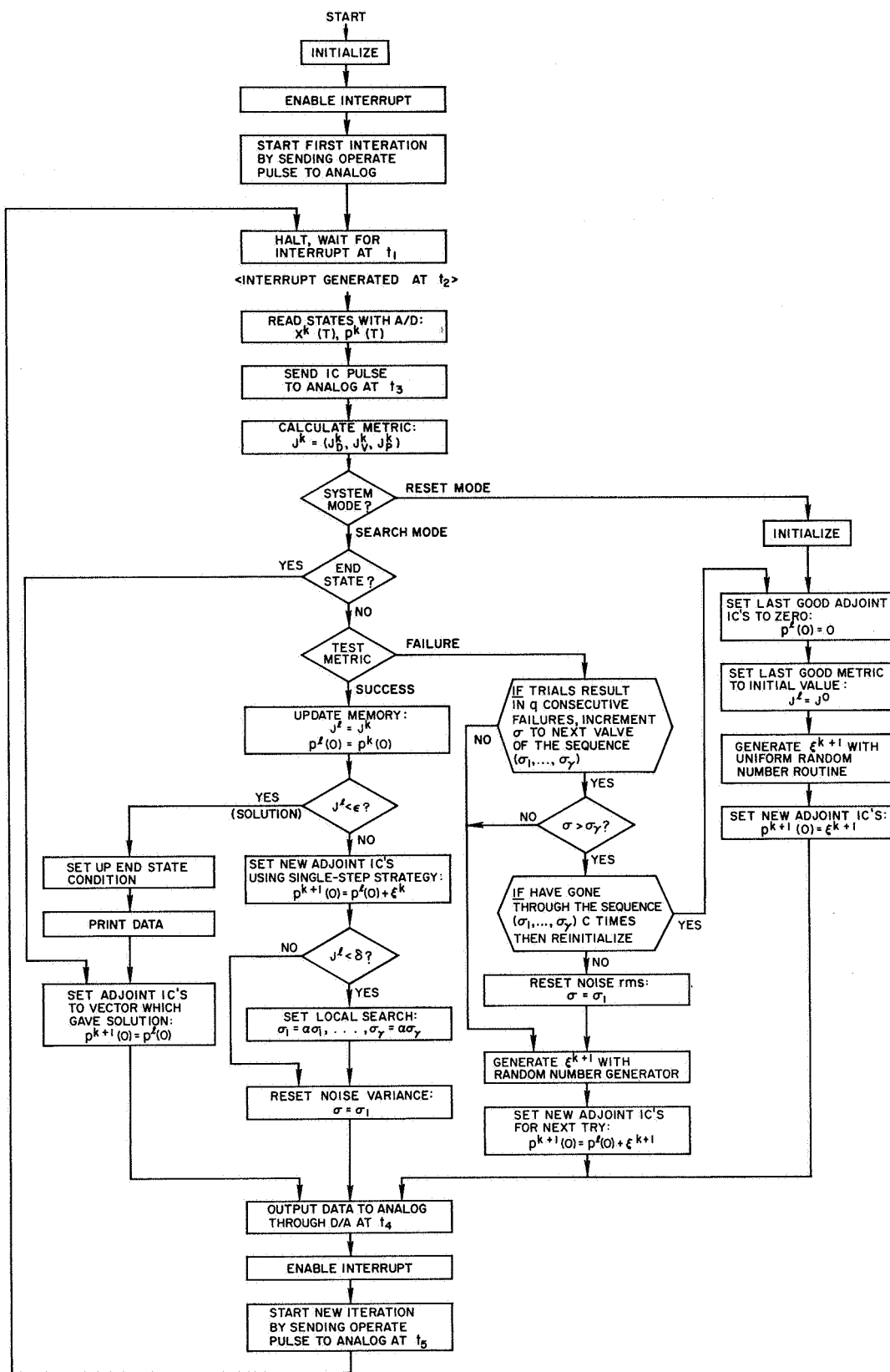


Figure 5.- Algorithm flow graph.

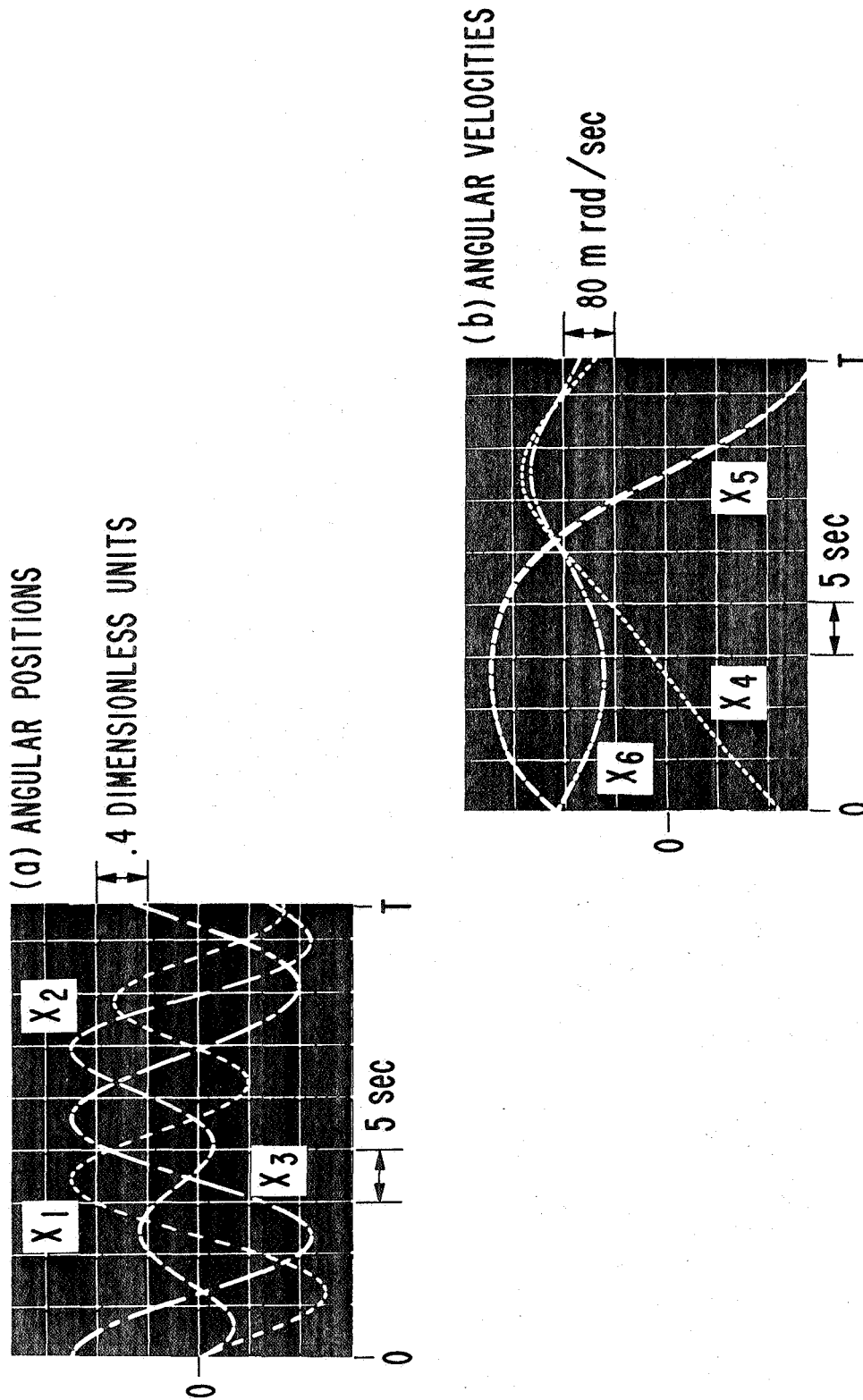


Figure 6. - Time history of state vector; no control;  
initial alignment of axes.

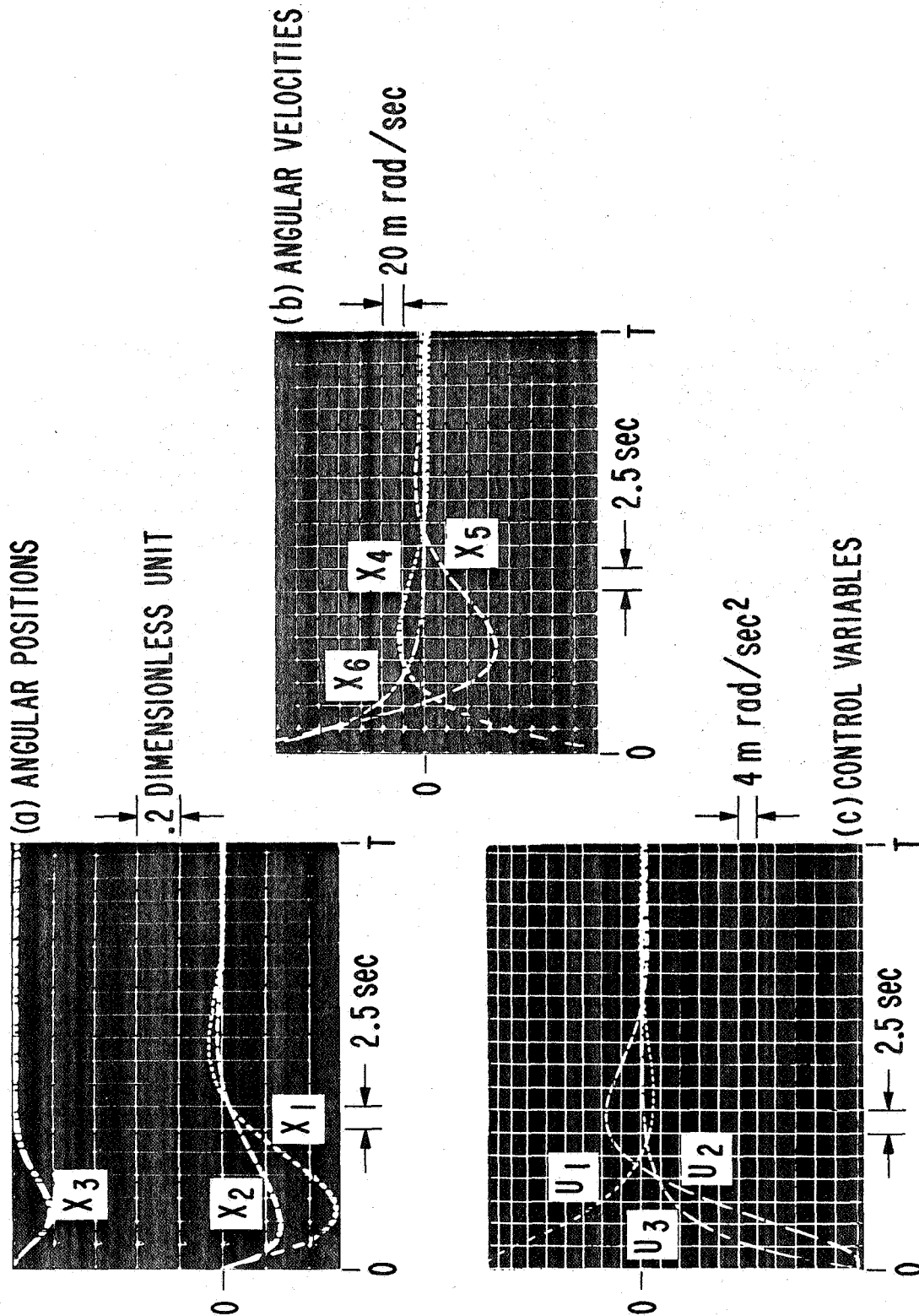


Figure 7. - Time histories of state and control vectors; optimal proportional control; initial alinement of axes.

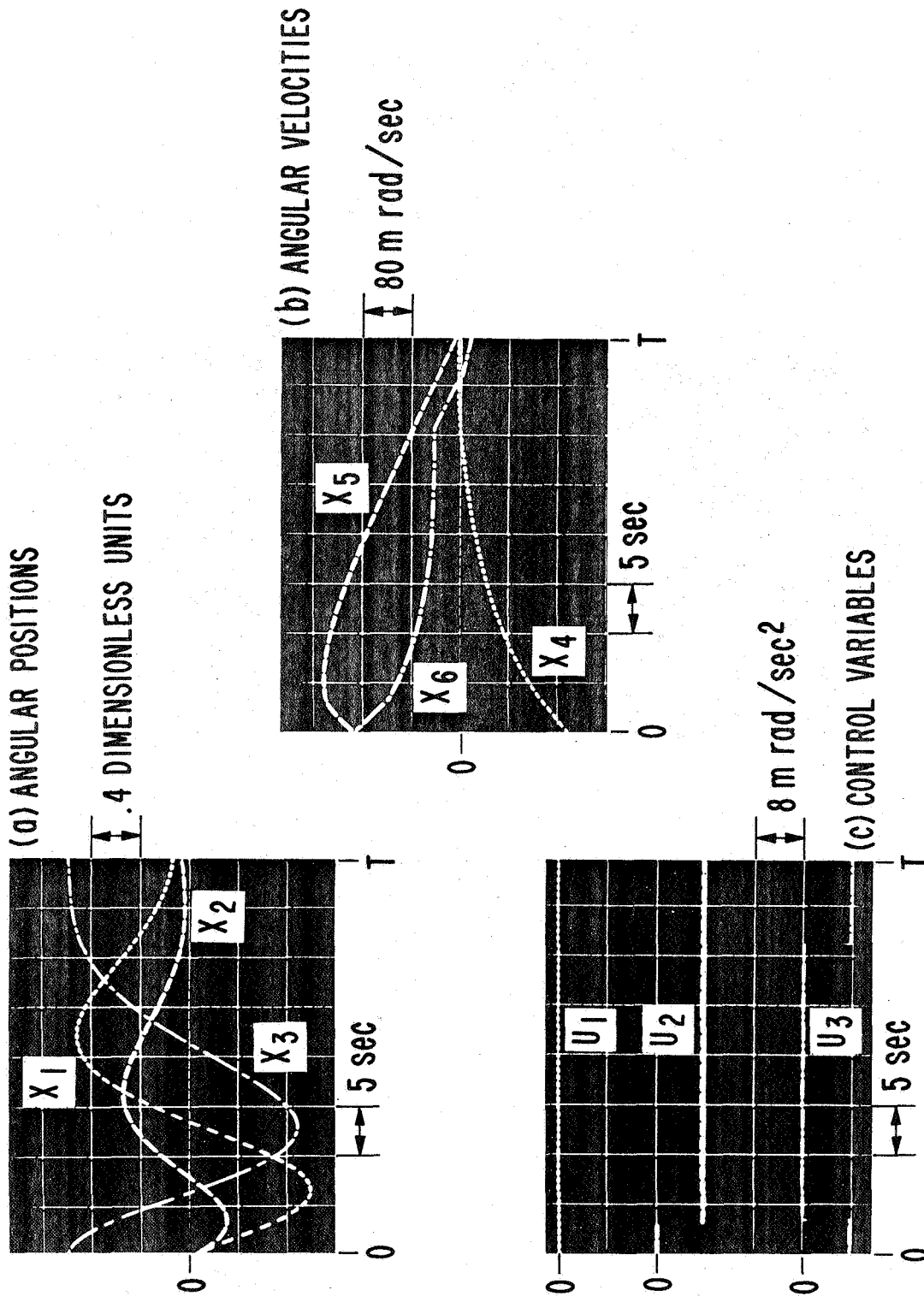


Figure 8. - Time histories of state and control vectors; optimal nonlinear control; initial alignment of axes.

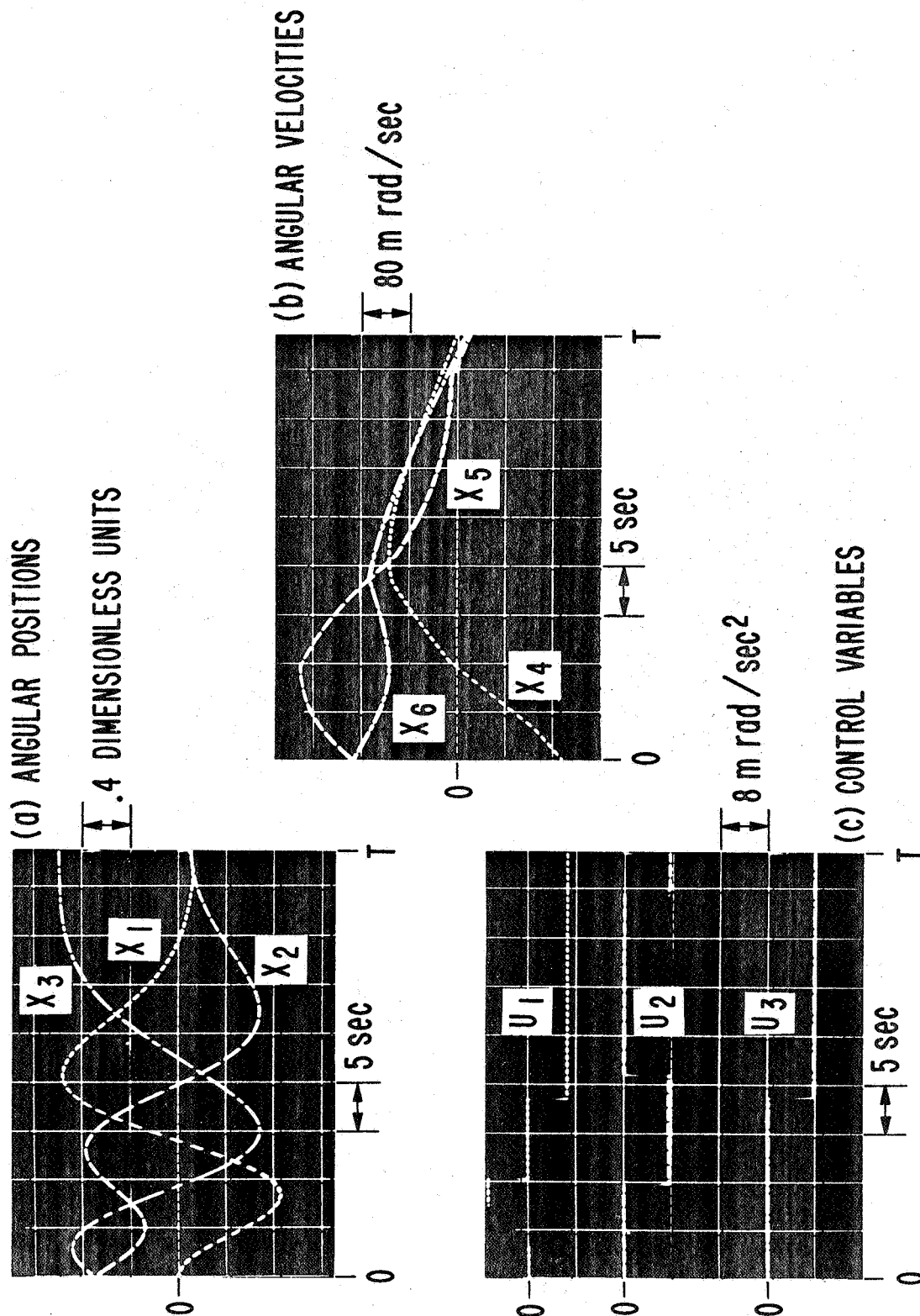


Figure 9. - Time histories of state and control vectors; optimal nonlinear control; initial nonalignment of axes.



SCALES  
 $J_D = 0.1$   
 DIMENSIONLESS  
 $J_V = 20 \text{ m rad/sec}$

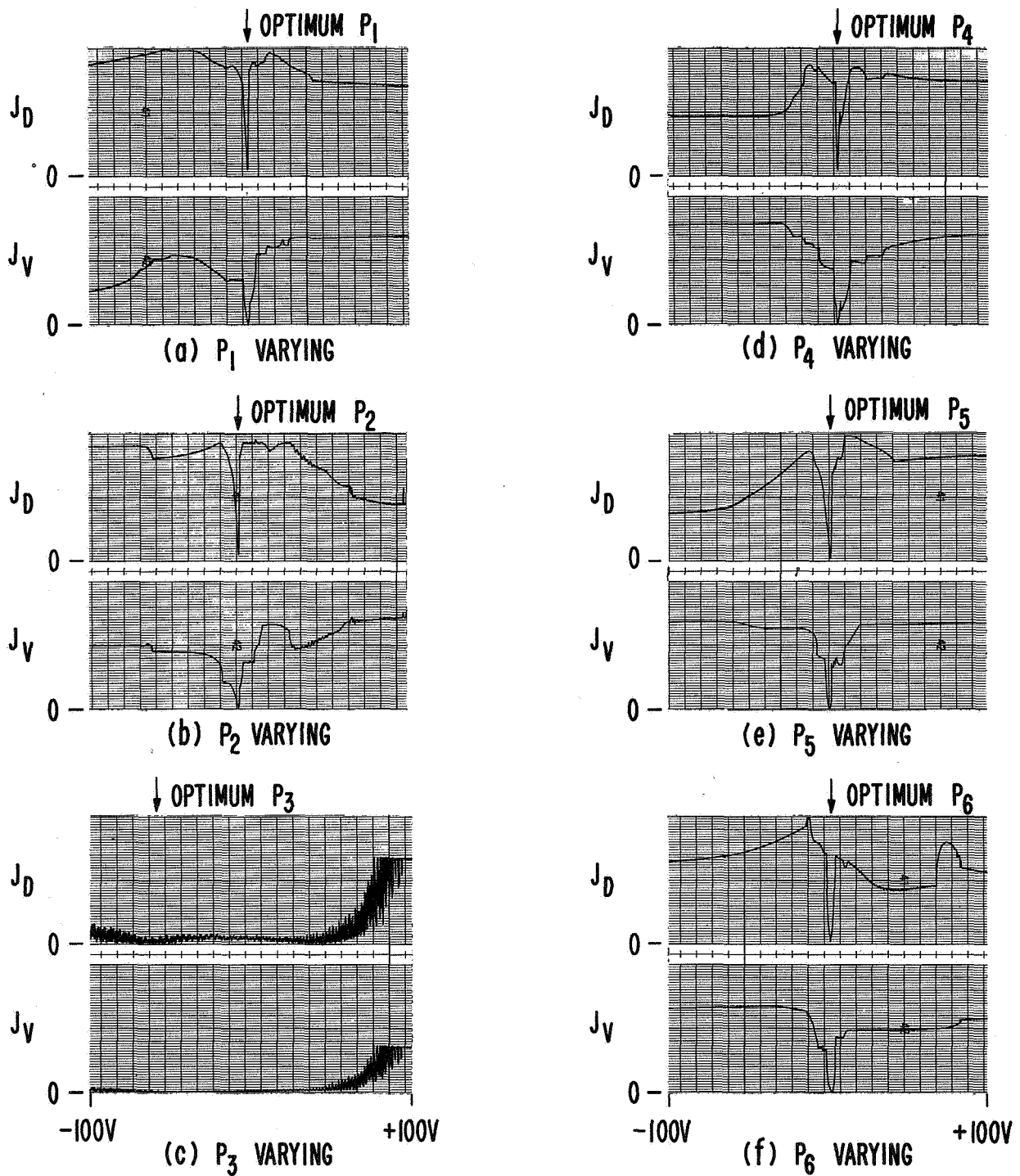


Figure 10.- Two-dimensional cross sections of boundary hypersurface; initial alinement of axes.